

## Claims

We claim:

- 5
1. A method for programmatically generating a second graphical program based on a first graphical program, the method comprising:
- receiving information specifying the first graphical program;
- programmatically generating the second graphical program based on the information, wherein the second graphical program implements functionality of the first graphical program.
- 10
2. The method of claim 1, wherein the second graphical program performs substantially like the first graphical program.
- 15
3. The method of claim 1, wherein the first graphical program implements first functionality; wherein the second graphical program also implements the first functionality.
- 20
4. The method of claim 1, wherein the second graphical program implements only a portion of functionality of the first graphical program.
- 25
5. The method of claim 1, wherein the first graphical program includes a first plurality of interconnected nodes which perform a first functionality; wherein the second graphical program includes a second plurality of interconnected nodes which perform the first functionality.

6. The method of claim 5, wherein the first plurality of interconnected nodes in the first graphical program visually indicate the first functionality of the first graphical program;

wherein the second plurality of interconnected nodes in the second graphical program visually indicate the first functionality of the second graphical program.

7. The method of claim 5,  
wherein the first plurality of interconnected nodes are interconnected according to one or more of data flow, control flow, and execution flow;  
wherein the second plurality of interconnected nodes are interconnected according to one or more of data flow, control flow, and execution flow.

8. The method of claim 1,  
wherein the first graphical program includes a first block diagram and a first user interface;  
wherein the second graphical program includes a second block diagram and a second user interface.

9. The method of claim 8,  
wherein the second block diagram appears substantially like the first block diagram.

10. The method of claim 8,  
wherein the second user interface appears substantially like the first user interface.

11. The method of claim 1,  
wherein the first graphical program is associated with a first programming development environment;

wherein the second graphical program is associated with a second programming development environment, wherein the second programming development environment is different than the first programming development environment.

5           12.    The method of claim 1,

wherein the first graphical program is developed according to a first programming development environment;

wherein the second graphical program is programmatically generated according to a second programming development environment, wherein the second programming development environment is different than the first programming development environment.

13.    The method of claim 1,

15       wherein the first graphical program is developed according to a first graphical programming language;

wherein the second graphical program is programmatically generated according to a second graphical programming language, wherein the second graphical programming language is different than the first graphical programming language.

20       14.    The method of claim 13,

wherein the first graphical programming language is the G language.

15.    The method of claim 13,

wherein the second graphical programming language is the G language.

25

16.    The method of claim 1,

wherein the first graphical program is one of a data flow program, control flow program, or an execution flow program;



24. The method of claim 1,  
wherein the received information specifying the first graphical program comprises  
a model file describing the first graphical program.

5

25. The method of claim 1, further comprising:  
constructing an abstract representation of the first graphical program based on the received information specifying the first graphical program;  
wherein said programmatically generating the second graphical program comprises programmatically creating the second graphical program based on the determined representation.

26. The method of claim 25,  
wherein said constructing an abstract representation of the first graphical program  
15 comprises constructing a directed graph representation of the first graphical program.

27. The method of claim 25,  
wherein said constructing an abstract representation of the first graphical program  
comprises constructing one or more data structures representing the first graphical  
20 program.

28. The method of claim 1,  
wherein the first graphical program includes a first node;  
wherein said receiving information specifying the first graphical program  
25 comprises receiving information specifying the first node;  
wherein said programmatically generating the second graphical program  
comprises programmatically including a second node corresponding to the first node in  
the second graphical program.



wherein the second graphical program includes a second plurality of interconnected nodes which perform the first functionality.

35. The memory medium of claim 34,  
5 wherein the first plurality of interconnected nodes are interconnected according to one or more of data flow, control flow, and execution flow;  
wherein the second plurality of interconnected nodes are interconnected according to one or more of data flow, control flow, and execution flow.

10 36. The memory medium of claim 30,  
wherein the first graphical program includes a first block diagram and a first user interface;  
wherein the second graphical program includes a second block diagram and a second user interface.

15 37. The memory medium of claim 30,  
wherein the first graphical program is associated with a first programming development environment;  
wherein the second graphical program is associated with a second programming  
20 development environment, wherein the second programming development environment is different than the first programming development environment.

25 38. A system for programmatically generating a second graphical program based on a first graphical program, the system comprising:  
a processor coupled to a memory, wherein the memory stores program instructions;  
wherein the processor is operable to execute the program instructions in order to:

receive information specifying the first graphical program;  
programmatically generate the second graphical program based on the  
information, wherein the second graphical program implements functionality of the first  
graphical program.

5

39. The system of claim 38,  
wherein the first graphical program implements first functionality;  
wherein the second graphical program also implements the first functionality.

10

40. The system of claim 38,  
wherein the second graphical program implements only a portion of functionality  
of the first graphical program.

15

41. The system of claim 38,  
wherein the first graphical program includes a first plurality of interconnected  
nodes which perform a first functionality;  
wherein the second graphical program includes a second plurality of  
interconnected nodes which perform the first functionality.

20

42. The system of claim 38,  
wherein the first plurality of interconnected nodes are interconnected according to  
one or more of data flow, control flow, and execution flow;  
wherein the second plurality of interconnected nodes are interconnected according  
to one or more of data flow, control flow, and execution flow.

25

43. The system of claim 38,  
wherein the first graphical program includes a first block diagram and a first user  
interface;



